Preprint

An Einstein-Bianchi system for Smooth Lattice General Relativity. II. 3+1 vacuum spacetimes.

Leo Brewin

School of Mathematical Sciences
Monash University, 3800
Australia

22-Mar-2011

Abstract

We will present a complete set of equations, in the form of an Einstein-Bianchi system, that describe the evolution of generic smooth lattices in spacetime. All 20 independent Riemann curvatures will be evolved in parallel with the leg-lengths of the lattice. We will show that the evolution equations for the curvatures forms a hyperbolic system and that the associated constraints are preserved. This work is a generalisation of our previous paper [1] on the Einstein-Bianchi system for the Schwarzschild spacetime to general 3+1 vacuum spacetimes.

1 Introduction

In a series of papers we have shown that the smooth lattice method works remarkably well for simple spacetimes such as the Schwarzschild spacetime in various slicings [1, 2], the maximally sliced Oppenheimer-Snyder spacetime [3], the vacuum Kasner cosmologies [4] and for constructing Schwarzschild initial data [5]. The equations are simple and require little computational sophistication to achieve stable and accurate results. The real test of the method however must be in the context of generic spacetimes. This paper is a first step in that direction.

The logic behind the smooth lattice approach is quite simple. We assume that we are given a lattice, built from a large collection of interconnected vertices, and where each path that connects a pair of vertices is taken to be a geodesic segment of the spacetime. The only data that we are given for the lattice is the connection matrix (which describes the topology as a list of pairs of connected vertices) and the lengths of each geodesic segment (which describes the metric properties). In this picture we are assuming that the lattice geometry is a close approximation to some underlying smooth geometry. The question that (should) spring to mind is – Given the leg lengths on a lattice, how do we compute the Riemann curvatures? We will return to this important question in just a moment, but for now let us suppose we have a suitable algorithm by which we can accurately compute the Riemann curvatures. It is then a simple matter to impose the vacuum Einstein equations* which in turn will impose constraints† on the leg-lengths. This furnishes us with a discrete set of equations for the leg-lengths. Solving these equations will yield a discrete solution of the vacuum Einstein equations.

We now return to the question of how to recover the Riemann curvatures given the set of leg-lengths. In one of our earlier papers [5] we argued that if the lattice was sufficiently well refined then a local Riemann normal coordinate frame could be constructed in the neighbourhood of any vertex extending to include, at least, the immediate neighbouring vertices. We called this neighbourhood the computational cell for the vertex (for lattices built from tetrahedra this would consist of the tetrahedra attached to the vertex). In this computational cell we can expand the metric as a power series [6] around the central vertex

$$g_{\mu\nu}(x) = g_{\mu\nu} - \frac{1}{3} R_{\mu\alpha\nu\beta} x^\alpha x^\beta + \mathcal{O}\left(L^3\right) \tag{1.1}$$

where L is a typical length scale for the computational cell. The requirement that the legs are geodesic segments leads, after some detailed calculations [6], to the following equation

$$L_{ij}^2 = g_{\mu\nu} \Delta x_{ij}^\mu \Delta x_{ij}^\nu - \frac{1}{3} R_{\mu\alpha\nu\beta} x_i^\mu x_i^\nu x_j^\alpha x_j^\beta + \mathcal{O}\left(L^5\right) \tag{1.2}$$

where $\Delta x_{ij}^\mu := x_j^\mu - x_i^\mu$. The approach advocated in [5] was to use this equation to extract the Riemann curvatures from the lattice. This may sound simple but there are a number of troubling issues.

*For pure pedagogy we will restrict the discussion to vacuum spacetimes.

†Not to be confused with any constraints that may exist at the continuum level, for example the ADM constraints.

The first issue concerns the coordinates. How do we compute coordinates for each vertex? Some can be set by simple gauge transformations (e.g., the origin can be tied to the central vertex) while the remainder must be computed from the lattice data (i.e., the leg-lengths). This forces us to view the above equations (1.2) as a coupled system for the curvatures and the coordinates.

The second issue is one of accountancy – do we have enough equations to compute the curvatures and the coordinates? For most lattices (in 3 and higher dimensions) the legs out number the coordinates x_i^μ and curvatures $R_{\mu\alpha\nu\beta}$. As an example, the computational cell used in our earlier paper [5] contained 78 legs and 19 vertices. Thus we had 78 equations for 6 curvatures and 57 coordinates (of which 6 can be freely chosen). There are at least two ways to handle this over supply of information. We can either form linear combinations of the above equations (1.2) to produce a reduced system in which the number of equations matches the number of unknowns. Or we can include a sufficient number of higher order terms in the Taylor series so as to produce a consistent set of equations. This later approach has the possible benefit of producing higher order approximations for the $R_{\mu\alpha\nu\beta}$ but at considerable extra expense. In both instances we still have a large coupled non-linear system of equations to solve at each vertex and at each time step. This is a considerable computational challenge.

Another important issue is one of uniqueness – how many distinct solutions can we find for the x_i^μ and $R_{\mu\alpha\nu\beta}$? The equations are non-linear and thus it is conceivable that more than one solution could be found. Do the solutions form a continuous family or are there only a finite set of solutions? How would we choose between these solutions? In our earlier paper [5] we resolved these problems by extending the lattice data to include the angles between each pair of legs attached to the central vertex. This allowed us to obtain an explicit and unique solution for all of the coordinates in a computational cell. It also had the added bonus of decoupling the coordinates from the curvatures – we could calculate all of the coordinates before computing the curvatures. The price we paid for this improvement was a significant increase in the number of data to be evolved. Where previously we had 78 legs per computational cell, now we had a further 33 angles.

However, there is a final issue which is much more serious than those just mentioned. To obtain $\mathcal{O}(L)$ accurate estimates for the curvatures, the coordinates must be computed to at least $\mathcal{O}(L^4)$ accuracy (i.e., the errors must be no worse than $\mathcal{O}(L^4)$). This follows by inspection of equation (1.2). Suppose the error in x_i^μ is $\mathcal{O}(L^a)$ for some $a > 0$. This error will couple with the first term on the right hand side of (1.2) to introduce an error of $\mathcal{O}(L^{a+1})$. But

the curvature terms are $\mathcal{O}(L^4)$ and will dominate the error term only when $a \geq 4$. Admittedly this is a somewhat naive analysis as it takes no account of the smoothness of the underlying geometry which might ensure that various lower order terms cancel (see for example the role smoothness plays in establishing the truncation errors in centred finite-difference approximations). But in the absence of an explicit algorithm we are unable to demonstrate that such cancellations do occur[‡]. The upshot is that if we persist with any of the variations suggested above we must design a solution strategy that guarantees, without invoking smoothness, that the errors in the coordinates are no worse than $\mathcal{O}(L^4)$. Despite our best efforts, we have not found a reliable solution to this problem.

These issues are not altogether new nor surprising and have proved to be a niggling concern throughout the development of the smooth lattice method. The only working solution that we have found (there may be others) is to surrender some (or all) of the main equations (1.2) in favour of the Bianchi identities. In all of our papers [1, 2, 3, 4, 5] we used a combination of the Bianchi identities and the geodesic deviation equation in $1+1$ spacetimes. The results were very encouraging. This was a hybrid scheme[§] and we attributed its success to the introduction of the Bianchi identities. This is the motivation for the present paper – Can we use the Bianchi identities to compute all of the Riemann curvatures in a $3+1$ spacetime? We should emphasise that there is one important difference between what we propose here and our previous work. In this paper we will use the full set of Bianchi identities to *evolve* all 20 independent Riemann curvatures. In contrast, in our $1+1$ experiments we used one Bianchi identity to compute one spatial curvature (i.e., a purely 3-dimensional computation within one Cauchy surface).

Why should we believe that this use of the Bianchi identities will overcome the issues described above? Simply, it allows us to use lower order approximations for the vertex coordinates (even flat space approximations) without compromising the quality of the estimates of the curvatures. We will return to this point after we have presented the full set of evolution equations.

[‡]Though the introduction of angles does produce an explicit algorithm its analysis is too unwieldily to be of any use.

[§]The geodesic deviation equation arises as a continuum limit of the smooth lattice equations [5].

2 Notation

A typical computational cell will be denoted by Ω. This will be a compact subset of the spacetime manifold. The central vertex of the cell will be denoted by O and the subset of Ω obtained by the intersection of Ω with the particular Cauchy surface that contains O will be denoted by ω. We will describe ω as the floor of Ω. As Ω has a finite extent there will be an image of ω that defines the future end of Ω. We will refer to this as the roof of Ω. We will have little to reason to refer to the past end of Ω but calling it the basement seems consistent.

We will assume throughout this paper that the vertex world lines are normal to the Cauchy surfaces (i.e., zero drift, in the language of [4]). This may seem restrictive but in our experiments to date it has worked very well.

Within Ω we will employ two sets of vectors essential to the evolution of the lattice. The first set will be an orthonormal tetrad, denoted by e_a, $a = 1, 2, 3, 4$, tied to the world line of O and aligned so that e_1 is the tangent vector to the world line of O. As we have assumed that the drift vector is everywhere zero this also ensures that e_1 is the future pointing unit normal to ω at O. Following convention, we will write n^μ as the unit normal to ω though as just noted, this is identical to e_1. The second set of vectors will be based on the set of radial legs attached to O. Each leg will be of the form (oi) and we will use v_i to denote the vector that joins (o) to (i). Note that the v_i are neither unit nor orthogonal. Latin characters will always be used to denote tetrad indices while the spacetime indices will be denoted by Greek letters. Latin characters will also be used as vertex labels and where confusion might arise we will use subsets of the Latin alphabet with $a, b, c, \cdots h$ reserved for frame components while i, j, k, l, m will be reserved for vertex labels. Obviously this distinction will only be imposed for equations that contain both types of index.

Each cell will carry a Riemann normal coordinate frame (an RNC frame), with coordinates $x^\mu = (t, x, y, z)$, tied to the central vertex and aligned with the tetrad. Note that this gives precedence to the tetrad over the coordinates. Coordinate components will be written as $R_{\mu\nu}$ or for specific components as, for example, R_{tx} while for frame components we will use scripts characters $\mathcal{R}_{ab}$. The coordinates for a typical vertex (i) will often be written as x^μ_i but on occasion we will have need to talk about the particular values for the x^μ_i in which case we will write $(t, x, y, z)_i$ or even x^t_i, x^z_i etc.

Each RNC frame will be chosen so that at O the metric is diagonal, $(g_{\mu\nu})_o =$

diag$(-1,1,1,1)$. Both spacetime and tetrad indices will be raised and lowered, at O, using the metric diag$(-1,1,1,1)$. With these choices we see that the future pointing unit normal to the Cauchy surface at the central vertex O is just $(n^\mu)_o = (1,0,0,0)^\mu$ while $(n_\mu)_o = (-1,0,0,0)$. We also see that the tetrad e_a has components ${e^\mu}_a = {\delta^\mu}_a$ in this RNC frame. Note that ${e^\mu}_a {e_\mu}^b = {\delta_a}^b$, ${e^\mu}_a {e^a}_\nu = {\delta^\mu}_\nu$, ${e^\mu}_1 = n^\mu$ and ${e_\mu}^1 = -n_\mu$.

3 Evolving the leg-lengths

The legs of the lattice are required to be short geodesic segments. Thus it should come as no surprise that the evolution of the leg-lengths can be obtained from the equations for the second variation of arc length. In an earlier paper [7] we showed that, for sufficiently short legs, these equations can be written as follows

$$\frac{dL_{ij}^2}{dt} = -2NK_{\mu\nu}\Delta x_{ij}^\mu \Delta x_{ij}^\nu + \mathcal{O}\left(L^3\right) \tag{3.1}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{1}{N}\frac{dL_{ij}^2}{dt}\right) &= 2N_{|\alpha\beta}\Delta x_{ij}^\alpha \Delta x_{ij}^\beta \\ &\quad + 2N\left(K_{\mu\alpha}K^\mu{}_\beta - R_{\mu\alpha\nu\beta}n^\mu n^\nu\right)\Delta x_{ij}^\alpha \Delta x_{ij}^\beta + \mathcal{O}\left(L^3\right)\end{aligned} \tag{3.2}$$

For numerical purposes it is somewhat easier to rewrite these in the following form

$$\frac{dL_{ij}^2}{dt} = -2NP_{ij} \tag{3.3}$$

$$\begin{aligned}\frac{dP_{ij}}{dt} &= -N_{|\alpha\beta}\Delta x_{ij}^\alpha \Delta x_{ij}^\beta \\ &\quad - N\left(K_{\mu\alpha}K^\mu{}_\beta - R_{\mu\alpha\nu\beta}n^\mu n^\nu\right)\Delta x_{ij}^\alpha \Delta x_{ij}^\beta\end{aligned} \tag{3.4}$$

in which we have introduced the new variables P_{ij}, one per leg. The $K_{\mu\nu}$ can be obtained by a suitable weighted sum of equation (3.1) as described in section (7.1). We have also dropped the truncation terms as these are not used during a numerical integration.

Clearly, the evolution of the leg lengths requires a knowledge of the Riemann curvatures and to that end we now present the evolution equations for those curvatures.

4 Evolving the Riemann curvatures. Pt. 1

We know that there are only 20 algebraically independent Riemann curvatures in 4 dimensions. So which should we choose? By a careful inspection of the algebraic symmetries of $R_{\mu\alpha\nu\beta}$ we settled upon the following

$$\begin{gathered} R_{xyxy},\ R_{xyxz},\ R_{xyyz},\ R_{xzxz},\ R_{xzyz},\ R_{yzyz} \\ R_{txxy},\ R_{txxz},\ R_{tyxy},\ R_{tyxz},\ R_{tyyz},\ R_{tzxy},\ R_{tzyz},\ R_{tzyz} \\ R_{txtx},\ R_{tyty},\ R_{tztz},\ R_{txty},\ R_{txtz},\ R_{tytz} \end{gathered} \tag{4.1}$$

4.1 Bianchi identities

Our aim is to use the Bianchi identities to obtain evolution equations for the Riemann curvatures. We begin by writing down the Bianchi identities at the central vertex, where the connection vanishes,

$$0 = R_{\mu\alpha\nu\beta,\gamma} + R_{\mu\alpha\beta\gamma,\nu} + R_{\mu\alpha\gamma\nu,\beta} \tag{4.2}$$

along with a contracted version of the same equation

$$0 = g^{\mu\gamma} R_{\mu\alpha\nu\beta,\gamma} - R_{\alpha\beta,\nu} + R_{\alpha\nu,\beta} \tag{4.3}$$

This pair of equations, along with the vacuum Einstein field equations, and a judicious choice of indices will provide us with all of the required evolution equations. This leads to the following 14 differential equations

$$0 = R_{xyxy,t} - R_{tyxy,x} + R_{txxy,y} \tag{4.4}$$
$$0 = R_{xyxz,t} - R_{tzxy,x} + R_{txxy,z} \tag{4.5}$$
$$0 = R_{xyyz,t} - R_{tzxy,y} + R_{tyxy,z} \tag{4.6}$$
$$0 = R_{xzxz,t} - R_{tzxz,x} + R_{txxz,z} \tag{4.7}$$
$$0 = R_{xzyz,t} - R_{tzxz,y} + R_{tyxz,z} \tag{4.8}$$
$$0 = R_{yzyz,t} - R_{tzyz,y} + R_{tyyz,z} \tag{4.9}$$
$$0 = R_{tyxy,t} - R_{xyxy,x} + R_{xyyz,z} \tag{4.10}$$
$$0 = R_{txxy,t} + R_{xyxy,y} + R_{xyxz,z} \tag{4.11}$$
$$0 = R_{tzxy,t} - R_{xyxz,x} - R_{xyyz,y} \tag{4.12}$$
$$0 = R_{tzxz,t} - R_{xzxz,x} - R_{xzyz,y} \tag{4.13}$$

$$0 = R_{txxz,t} + R_{xyxz,y} + R_{xzxz,z} \tag{4.14}$$
$$0 = R_{tyxz,t} - R_{xyxz,x} + R_{xzyz,z} \tag{4.15}$$
$$0 = R_{tzyz,t} - R_{xzyz,x} - R_{yzyz,y} \tag{4.16}$$
$$0 = R_{tyyz,t} - R_{xyyz,x} + R_{yzyz,z} \tag{4.17}$$

There are of course 20 independent $R_{\mu\alpha\nu\beta}$, 14 of which are subject to the above evolution equations while the remaining 6 can be obtained from the vacuum Einstein equations

$$0 = R_{xx} = -R_{txtx} + R_{xyxy} + R_{xzxz} \tag{4.18}$$
$$0 = R_{yy} = -R_{tyty} + R_{xyxy} + R_{yzyz} \tag{4.19}$$
$$0 = R_{zz} = -R_{tztz} + R_{xzxz} + R_{yzyz} \tag{4.20}$$
$$0 = R_{xy} = -R_{txty} + R_{xzyz} \tag{4.21}$$
$$0 = R_{xz} = -R_{txtz} - R_{xyyz} \tag{4.22}$$
$$0 = R_{yz} = -R_{tytz} + R_{xyxz} \tag{4.23}$$

Though these are not differential equations they do, none the less, provide a means to evolve the 6 curvatures $R_{txtx}, R_{txty} \cdots R_{tytz}$.

The important point to note about this system of equations is that it is closed, there are 20 evolution equations for 20 curvatures. The source terms, such as $R_{xyxy,x}$, could be computed by importing data from the neighbouring cells, by an appropriate combination of rotations and boosts, and using a suitable finite difference approximation (see section (7) for more details)). In this way the lattice serves as a scaffold on which source terms such as these can be computed.

4.2 Constraints

In deriving the 20 evolution equations of the previous section we used only 6 of the 10 vacuum Einstein equations. Thus the 4 remaining vacuum Einstein equations must be viewed as constraints. These equations are

$$0 = R_{tt} = \quad R_{txtx} + R_{tyty} + R_{tztz} \tag{4.24}$$
$$0 = R_{tx} = \quad R_{tyxy} + R_{tzxz} \tag{4.25}$$
$$0 = R_{ty} = -R_{txxy} + R_{tzyz} \tag{4.26}$$
$$0 = R_{tz} = -R_{txxz} - R_{tyyz} \tag{4.27}$$

Finally, we have the following 6 constraints that arise from the Bianchi identities.

$$0 = R_{xyxy,z} + R_{xyyz,x} - R_{xyxz,y} \tag{4.28}$$
$$0 = R_{xyxz,z} + R_{xzyz,x} - R_{xzxz,y} \tag{4.29}$$
$$0 = R_{xyyz,z} + R_{yzyz,x} - R_{xzyz,y} \tag{4.30}$$
$$0 = R_{txxy,z} + R_{txyz,x} - R_{txxz,y} \tag{4.31}$$
$$0 = R_{tyxy,z} + R_{tyyz,x} - R_{tyxz,y} \tag{4.32}$$
$$0 = R_{tzxy,z} + R_{tzyz,x} - R_{tzxz,y} \tag{4.33}$$

So all up we have 20 evolution equations assembled from the 14 differential equations (4.4–4.17) and 6 algebraic equations (4.18–4.23) plus 10 constraints comprising 4 Einstein equations (4.24–4.27) and 6 Bianchi identities (4.28–4.33). This is a such a simple system that it allows simple questions to be explored and answered with ease. The questions that we will address are

1. Are the constraints preserved by the evolution equations?
2. Do the evolution equations constitute a hyperbolic system?

For both questions the answer is yes and we shall now demonstrate that this is so.

4.3 Constraint preservation

In the following discussion we will assume that, by some means, we have constructed an initial data set for the 20 $R_{\mu\alpha\nu\beta}$. That is, the 20 $R_{\mu\alpha\nu\beta}$ are chosen so that the 10 constraints (4.24–4.33) vanish at the central vertex of *every* computational cell in the lattice.

We will also need the trivial result that

$$R = 2\left(R_{xyxy} + R_{xzxz} + R_{yzyz}\right) \tag{4.34}$$

which follows directly from equations (4.18,4.19,4.20,4.24).

Consider now the constraint $0 = R_{tz}$. By assumption, this constraint is satisfied on the initial slice. To demonstrate that it continues to hold throughout the evolution we need to show that $0 = R_{tz,t}$. From (4.27) this requires us to show that $0 = R_{txxz,t} + R_{tyyz,t}$. Using (4.14,4.17) we see that

$$R_{txxz,t} + R_{tyyz,t} = -R_{xyxz,y} - R_{xzxz,z} + R_{xyyz,x} - R_{yzyz,z}$$

however on the initial slice we also have, by assumption, (4.28)

$$0 = R_{xyxy,z} + R_{xyyz,x} - R_{xyxz,y}$$

which when combined with the previous equation leads to

$$R_{txxz,t} + R_{tyyz,t} = -\left(R_{xyxy} + R_{xzxz} + R_{yzyz}\right)_{,z}$$

But by equation (4.34) we see that the right hand side is just $-R_{,z}/2$ and as $R = 0$ across the initial slice we also have that $0 = R_{,z}$ at every central vertex. This completes the proof. The two other constraints, $0 = R_{ty}$ and $0 = R_{tx}$, can be dealt with in a similar fashion.

All that remains is to show that $0 = R_{tt}$ is conserved. We proceed in a manner similar to the above. First we use $R_{tt} = R_{xyxy} + R_{xzxz} + R_{yzyz}$ and then use equations (4.4,4.7,4.9) to compute the time derivative

$$\begin{aligned}
\left(R_{xyxy} + R_{xzxz} + R_{yzyz}\right)_{,t} &= R_{xyxy,t} + R_{xzxz,t} + R_{yzyz,t} \\
&= \quad R_{tyxy,x} - R_{txxy,y} \\
&\quad + R_{tzxz,x} - R_{txxz,z} \\
&\quad + R_{tzyz,y} - R_{tyyz,z} \\
&= R_{tx,x} + R_{ty,y} + R_{tz,z}
\end{aligned}$$

where the last line arose by inspection of equations (4.25–4.27). But $0 = R_{\mu\nu}$ at every central vertex on the initial slice. Thus $0 = R_{\mu\nu,i}$, $i = x, y, z$ on the central vertex which in turn shows that $0 = R_{tt,t}$ on the initial slice.

A key element in the above proofs was the use of constraints based on the Bianchi identities. The question now must be – do the evolution equations preserve those constraints? The answer is yes which we will now demonstrate on a typical case. Consider the constraint (4.28)

$$0 = R_{xyxy,z} + R_{xyyz,x} - R_{xyxz,y}$$

We know this to be true on the initial slice and we need to show that the evolution equations (4.4–4.17) guarantee that it will be satisfied on all subsequent slices. The calculations follow a now familiar pattern,

$$\begin{aligned}
\left(R_{xyxy,z} + R_{xyyz,x} - R_{xyxz,y}\right)_{,t} &= R_{xyxy,tz} + R_{xyyz,tx} - R_{xyxz,ty} \\
&= \quad \left(R_{tyxy,x} - R_{txxy,y}\right)_{,z} \\
&\quad + \left(R_{tzxy,y} - R_{tyxy,z}\right)_{,x} \\
&\quad - \left(R_{tzxy,x} - R_{txxy,z}\right)_{,y} \\
&= 0
\end{aligned}$$

The same analysis can be applied to the remaining constraint equations.

4.4 Hyperbolicity

Our approach to proving hyperbolicity will be quite simple. We will manipulate the evolution equations (4.4–4.17) to demonstrate that each of our 20 $R_{\mu\alpha\nu\beta}$ satisfies the standard second order wave equation.

Let us start with a simple example, equation (4.4). We take one further time derivative, commute the mixed partial derivatives and then use equations (4.10,4.11) to eliminate the single time derivative. This leads to

$$0 = R_{xyxy,tt} - R_{xyxy,xx} - R_{xyxy,yy} + R_{xyyz,zx} - R_{xyxz,zy}$$

However, we also have $0 = R_{xyxy,z} + R_{xyyz,x} - R_{xyxz,y}$, which allows us to reduce the last two terms of the previous equation to just $-R_{xyxy,zz}$. Thus we have

$$0 = R_{xyxy,tt} - R_{xyxy,xx} - R_{xyxy,yy} - R_{xyxy,zz}$$

This is the standard flat space wave equation for R_{xyxy}. A similar analysis shows that R_{xzxz}, R_{yzyz}, R_{xyxz}, R_{xyyz} and R_{xzyz} are also solutions of the wave equation.

We now turn to the 8 $R_{\mu\alpha\nu\beta}$ in which the indices $\mu\alpha\nu\beta$ contain just one t. The proof (that each such $R_{\mu\alpha\nu\beta}$ satisfies the wave equation) differs from the above only in the way the Bianchi identities are used. Applying the first few steps outlined above to equation (4.10) leads to

$$\begin{aligned} 0 = R_{tyxy,tt} - R_{tyxy,xx} - R_{tyxy,yy} - R_{tyxy,zz} \\ + R_{tyxy,yy} + R_{txxy,xy} + R_{tzxy,zy} \end{aligned}$$

in which we have deliberately introduced the pair of terms $R_{tyxy,yy}$ to aid in the following exposition. The last three terms can be dealt with as follows. First notice that

$$\begin{aligned} R_{tyxy,yy} + R_{txxy,xy} + R_{tzxy,zy} &= \left(R_{tyxy,y} + R_{txxy,x} + R_{tzxy,z}\right)_{,y} \\ &= \left(-R^{\mu}{}_{txy,\mu}\right)_{,y} \\ &= \left(-R_{ty,x} + R_{tx,y}\right)_{,y} \end{aligned}$$

where in last line we have used the contracted Bianchi identity $0 = R^{\mu}{}_{\nu\alpha\beta,\mu} - R_{\nu\beta,\alpha} + R_{\nu\alpha,\beta}$. But we know that $0 = R_{\mu\nu}$ at every central vertex, thus all of its partial derivatives will be zero and so the each term on the right hand vanishes leading to our desired result

$$0 = R_{tyxy,tt} - R_{tyxy,xx} - R_{tyxy,yy} - R_{tyxy,zz}$$

Finally we note that the remaining 6 $R_{\mu\alpha\nu\beta}$, that is those that carry two t's in their indices, are linear combinations of the previous 14 $R_{\mu\alpha\nu\beta}$, see equations (4.18–4.23), and thus will also be solutions of the wave equation. Thus we have shown, as claimed, that all 20 $R_{\mu\alpha\nu\beta}$ satisfy the wave equation.

5 Evolving the Riemann curvatures. Pt. 2

There are two problems in the forgoing analysis. The first problem is that we chose a unit lapse function when presenting the evolution equations (4.4–4.17). We can easily remedy this problem by making a simple vertex dependent coordinate substitution $t = Nt'$ in each of the evolution equations.

The second problem is somewhat more of a challenge. It stems from the simple fact that each computational cell is local in both space and time and therefore no single RNC can be used to track the evolution for an extended period of time. We will have no choice but to jump periodically to a new RNC frame. But how might we do this? One approach goes as follows. Build, on the world line of a typical vertex, a pair of distinct but overlapping cells, with one cell lying slightly to the future of the other. Then evolve the curvatures in the frame of one cell into the overlap region followed by a coordinate transformation to import the newly evolved curvatures into the frame of the future cell. This completes one time step of the integration whereupon the whole process can be repeated any number of times along the vertex world line. A useful improvement on this is to use a local tetrad to construct scalars thus avoiding the need for explicit coordinate transformations when passing from one cell to the next. The price we pay for this is that we have to account for the evolution of the tetrad along the world line. As we shall see this is rather easy to do (essentially we project the tetrad onto the legs of the lattice). We will explore this method first on a simple example before presenting the computations for the curvature evolution equations.

5.1 A simple example

In this example we will suppose that we have a vector W^μ that evolves along the world line of the central vertex according to

$$\frac{dW^\mu}{dt'} = NF^\mu \tag{5.1}$$

Our aim is to obtain a related equation that describes the evolution of the vector along the whole length of the world line, not just the short section contained within this one cell.

Suppose that we have an orthonormal tetrad $e_a = e^\mu{}_a \partial_\mu$, $a = 1, 2, 3, 4$ on ω with e_1 aligned to $n^\mu \partial_\mu$, the future pointing normal to ω, and that we have aligned the RNC coordinate axes with the tetrad (note how this gives precedence to the tetrad over the coordinates). Thus at the central vertex of Ω we have

$$
\begin{gathered}
e_a = \partial_a \,, \quad e^\mu{}_a = \delta^\mu{}_a \,, \quad e_\mu{}^a = \delta_\mu{}^a \\
n^\mu = e^\mu{}_1 \,, \quad -n_\mu = e_\mu{}^1 \\
e^\mu{}_a e_\mu{}^b = \delta_a{}^b \,, \quad e^\mu{}_a e_\nu{}^a = \delta_\mu{}^\nu \\
g_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1) \,, \quad g_{ab} = \mathrm{diag}(-1, 1, 1, 1)
\end{gathered}
$$

We now propose the following evolution equations along the world line of the central vertex in Ω.

$$
\frac{de^\mu{}_1}{dt'} = e^\mu{}_i \nabla^i N \,, \quad \frac{de_\mu{}^1}{dt'} = -e_\mu{}^i \nabla_i N \tag{5.2}
$$

$$
\frac{de^\mu{}_i}{dt'} = e^\mu{}_1 \nabla_i N \,, \quad \frac{de_\mu{}^i}{dt'} = -e_\mu{}^1 \nabla^i N \,, \quad i = 2, 3, 4 \tag{5.3}
$$

where $\nabla_i N = (\perp N_{,\nu}) e^\nu{}_i$ and $\nabla^i N = (\perp N^{,\nu}) e_\nu{}^i$, $i = 2, 3, 4$. What can we say about the evolved data? First, note that the orthonormal conditions are preserved, that is

$$
\frac{de^\mu{}_a e_\mu{}^b}{dt'} = 0 \,, \quad \frac{de^\mu{}_a e_\nu{}^a}{dt'} = 0
$$

Thus the tetrad obtained by integrating the above equations will remain orthonormal along the world line of the central vertex. Second, using

$$
(N n^\mu)_{;\nu} = N_{,\nu} n^\mu - \perp(N^{,\mu}) n_\nu - N K^\mu{}_\nu \tag{5.4}
$$

to compute $dn^\mu/dt' = n^\mu{}_{;\nu} \, (N n^\nu)$ we see that

$$
\frac{de^\mu{}_1}{dt'} = \frac{dn^\mu}{dt'} \,, \quad \frac{de_\mu{}^i}{dt'} = -\frac{dn_\mu}{dt'} \tag{5.5}
$$

which shows that $e^\mu{}_1 = n^\mu$ and $e_\mu{}^1 = -n_\mu$ everywhere along the world line. That is, $e^\mu{}_1$ remains tied to the world line. All that remains is to account for how the tetrad rotates around the world line. This we shall do by evolving the projections of the $e^\mu{}_i$, $i = 2, 3, 4$ onto the legs of the lattice. Let $v_a = v^\mu{}_a \partial_\mu$,

$a = 1, 2, 3$ be any three distinct legs of the lattice attached to the central vertex. Now consider a short time step in which the vector v_a sweeps out a short quadrilateral in spacetime (see figure (2)). The upper and lower edges will be the past and future versions of v_a while the remaining two sides will be generated by the word lines of the vertices that define v_a. Since we have assumed at the outset that all vertices evolve normal to the Cauchy surface we see that these vertical vectors correspond to Nn^μ. The important point is that this set of four vectors forms a closed loop, in short the vectors v_a and $Nn^\mu\partial_\mu$ commute, thus

$$v^\mu{}_{a;\nu}\left(Nn^\nu\right) = v^\nu{}_a\left(Nn^\mu\right)_{;\nu} \tag{5.6}$$

The left hand side is simply dv^μ/dt', while the right hand side can be expanded using (5.4). This leads to

$$\frac{dv^\mu{}_a}{dt'} = \left(N_{,\nu}n^\mu - NK^\mu{}_\nu\right)v^\nu{}_a \tag{5.7}$$

where we have dropped the term involving $n_\mu v^\mu{}_a$ as this would be $\mathcal{O}\left(L^m\right)$ with $m \geq 2$ while the remaining terms are all $\mathcal{O}\left(L\right)$.

We are now ready to construct our scalar evolution equations. Let $\mathcal{W}_a := W_\mu e^\mu{}_a$ and $\mathrm{v}_a{}^b := v^\mu{}_a e_\mu{}^b$ then

$$\frac{d\mathcal{W}_a}{dt} = \frac{dW_\mu}{dt'}e^\mu{}_a + W_\mu\frac{de^\mu{}_a}{dt'}$$

$$\frac{d\mathrm{v}_a{}^b}{dt'} = \frac{dv^\mu{}_a}{dt'}e_\mu{}^b + v^\mu{}_a\frac{de_\mu{}^b}{dt'}\,, \quad b = 1, 2, 3, 4$$

Each of these equations can be re-cast entirely in terms of the scalars by first using (5.2,5.3,5.7) to eliminate the time derivatives on the right hand side followed by the substitutions $W_\mu = \mathcal{W}_a e_\mu{}^a$ and $v^\mu{}_a = \mathrm{v}_a{}^b e^\mu{}_b$. This leads to

$$\frac{d\mathcal{W}_n}{dt'} = N\mathcal{F}_n + \mathcal{W}_i\nabla^i N \tag{5.8}$$

$$\frac{d\mathcal{W}_i}{dt'} = N\mathcal{F}_i + \mathcal{W}_n\nabla_i N\,, \quad i = 2, 3, 4 \tag{5.9}$$

$$\frac{d\mathrm{v}_a{}^1}{dt'} = \frac{1}{N}\frac{dN}{dt'}\mathrm{v}_a{}^1 \tag{5.10}$$

$$\frac{d\mathrm{v}_a{}^i}{dt'} = -N\mathcal{K}^i{}_j\mathrm{v}_a{}^j\,, \quad i, j = 2, 3, 4\,, \;\; a = 1, 2, 3 \tag{5.11}$$

where we have introduced the scalars $\mathcal{W}_n = W_\mu n^\mu$, $\mathcal{F}_n = F_\mu n^\mu$, $\mathcal{F}_i = F_\mu e^\mu{}_i$, and $\mathcal{K}^i{}_j = K^\mu{}_\nu e_\mu{}^i e^\nu{}_j$. These are our final equations. They are valid along the whole length of the world line, not just the part contained in one cell.

Equation (5.11) describes the motion of the tetrad relative to the legs of the lattice. As we integrate forward in time we can use the values of $v_a{}^i$ to locate the tetrad within the computational cell. If we chose to construct an RNC within the cell then we can go one step further and recover the values of $e^\mu{}_i$ and the W^μ.

5.2 Curvature evolution equations

Now we can return to the task of constructing the generalised evolution equations for the curvatures. We start by introducing a pair of relations between the tetrad and coordinate components of the curvature tensor

$$\mathcal{R}_{abcd} = R_{\mu\alpha\nu\beta} e^\mu{}_a e^\alpha{}_b e^\nu{}_c e^\beta{}_d$$
$$R_{\mu\alpha\nu\beta} = \mathcal{R}_{abcd} e_\mu{}^a e_\alpha{}^b e_\nu{}^c e_\beta{}^d$$

and then forming a typical evolution equation

$$\frac{d\mathcal{R}_{abcd}}{dt'} = \frac{dR_{\mu\alpha\nu\beta}}{dt'} e^\mu{}_a e^\alpha{}_b e^\nu{}_c e^\beta{}_d + R_{\mu\alpha\nu\beta} \frac{d\left(e^\mu{}_a e^\alpha{}_b e^\nu{}_c e^\beta{}_d\right)}{dt'} \tag{5.12}$$

with each d/dt' term on the right hand side replaced by a suitable combination of the existing evolution equations, (4.4–4.17) for the curvature terms and (5.2,5.3) for the tetrad terms.

Rather than working through all 14 equations we will demonstrate the procedure on just one equation (4.4) leaving the remaining equations (but not their working) to the Appendix. So our starting point is

$$\frac{d\mathcal{R}_{xyxy}}{dt'} = \frac{dR_{xyxy}}{dt'} + R_{\mu\alpha\nu\beta} \frac{d\left(e^\mu{}_x e^\alpha{}_y e^\nu{}_x e^\beta{}_y\right)}{dt'}$$

and using (4.4) we obtain

$$\frac{d\mathcal{R}_{xyxy}}{dt'} = R_{tyxy,x} - R_{txxy,y} + R_{\mu\alpha\nu\beta} \frac{d\left(e^\mu{}_x e^\alpha{}_y e^\nu{}_x e^\beta{}_y\right)}{dt'}$$

Finally we use (5.2,5.3) to eliminate the time derivative of $e^\mu{}_a$, leading to

$$\begin{aligned}\frac{d\mathcal{R}_{xyxy}}{dt'} &= R_{tyxy,x} - R_{txxy,y} \\ &\quad + \mathcal{R}_{tyxy}\nabla_x N - \mathcal{R}_{txxy}\nabla_y N + \mathcal{R}_{tyxy}\nabla_x N - \mathcal{R}_{txxy}\nabla_y N\end{aligned} \tag{5.13}$$

This is as far as we need go, though it is tempting to make the substitutions $R_{tyxy} = \mathcal{R}_{abcd} e_t{}^a e_y{}^b e_x{}^c e_y{}^d$ and $R_{txxy} = \mathcal{R}_{abcd} e_t{}^a e_x{}^b e_x{}^c e_y{}^d$. But that is not really necessary as we can defer those substitutions until we actually need values for the stated partial derivatives. This is described in more detail in section (7).

Note that when introducing the lapse function by the substitution $t = Nt'$ we have not made explicit the coordinate transformation on the curvatures other than to use distinct labels t and t'. In this way we use t' as an integration parameter on the world line of each vertex while retaining the original coordinates (t, x, y, z) as the local Riemann normal coordinates (and thus at any point on the world line we continue to have $(g_{\mu\nu})_o = \text{diag}(-1, 1, 1, 1)$). We choose to maintain this distinction between t and t' not only to keep the equations tidy but also because it leaves the equations in a simple form well suited to numerical integrations.

Clearly the above procedure can be applied directly to each of the remaining 13 curvature evolution equations. The final results for all 14 equations can be found in the Appendix.

5.3 Hyperbolicity and constraint preservation

It is natural to ask if the new system of evolution equations are hyperbolic and also, are the new constraints preserved by the new evolution equations? The answer to both questions is yes and we will demonstrate this as follows.

Given that $\mathcal{R}_{abcd} = R_{\mu\alpha\nu\beta} e^\mu{}_a e^\alpha{}_b e^\nu{}_c e^\beta{}_d$ we see that

$$\mathcal{R}_{abcd,ef} = R_{\mu\alpha\nu\beta,\rho\tau} e^\mu{}_a e^\alpha{}_b e^\nu{}_c e^\beta{}_d e^\rho{}_e e^\tau{}_f + \mathcal{V}_{abcdef}\left(R, N, \partial R, \partial N, \partial^2 N\right)$$

where $\mathcal{V}_{abcdef}$ is a function of $R_{\mu\alpha\nu\beta}$, N and the indicated partial derivatives. Importantly, $\mathcal{V}_{abcdef}$ does not contain any second partial derivatives of the curvatures. We have previously shown that, at the central vertex, each $R_{\mu\alpha\nu\beta}$ satisfies a wave equation of the form $0 = g^{\rho\tau} R_{\mu\alpha\nu\beta,\rho\tau}$ with $g^{\rho\tau} = \text{diag}(-1, 1, 1, 1)$. Thus we find that

$$g^{ef}\mathcal{R}_{abcd,ef} = g^{ef}\mathcal{V}_{abcdef}\left(R, \partial R, N, \partial N, \partial^2 N\right)$$

where $g^{ef} = \text{diag}(-1, 1, 1, 1)$. It follows that each $\mathcal{R}_{abcd}$ satisfies a wave equation with source terms and therefore we have shown that the new evolution equations constitute a hyperbolic system.

A similar analysis can be applied to the constraints. We begin by writing a typical differential constraint (4.28–4.33) in the form

$$0 = W_{\mu\alpha\nu\beta}(\partial R)$$

where the right hand side depends only on the the first derivatives of $R_{\mu\alpha\nu\beta}$. Introducing the lapse function is trivial (there are no time derivatives, so the equation is unchanged). If we define the frame components $\mathcal{W}_{abcd}$ by

$$\mathcal{W}_{abcd} = W_{\mu\alpha\nu\beta} e^{\mu}{}_{a} e^{\alpha}{}_{b} e^{\nu}{}_{c} e^{\beta}{}_{d}$$

then we find

$$\mathcal{W}_{abcd,t} = W_{\mu\alpha\nu\beta,\rho} e^{\mu}{}_{a} e^{\alpha}{}_{b} e^{\nu}{}_{c} e^{\beta}{}_{d} e^{\rho}{}_{t} + W_{\mu\alpha\nu\beta} \left(e^{\mu}{}_{a} e^{\alpha}{}_{b} e^{\nu}{}_{c} e^{\beta}{}_{d} \right)_{,t}$$

and as we have previously shown that $W_{\mu\alpha\nu\beta} = 0$ and $W_{\mu\alpha\nu\beta,\rho} = 0$ it follows that $\mathcal{W}_{abcd} = 0$ and $\mathcal{W}_{abcd,t} = 0$. It is easy to see that the same procedure can be applied to the remaining constraints (4.18–4.24) with the same outcome. Thus we have shown that the new constraints are conserved by the new evolution equations.

6 Coordinates

There are at least two instances where the vertex coordinates are required. First, when constructing the transformation matrix used when importing data from neighbouring cells. Second, as part of the time integration of leg-lengths, equations (3.1–3.2). They are also required when computing the extrinsic curvatures (7.1) and the hessian (7.2).

Recall that within each cell we employ two distinct coordinate frames, one is tied to the tetrad associated with the central vertex while the other is aligned with the lattice. Both frames share the central vertex as the origin. We will describe first how to construct the lattice coordinates, which we will denote by y^{μ}, followed by the tetrad coordinates, denoted by x^{μ}. The lattice coordinates are only ever used in the construction of the tetrad coordinates, once these are known then the lattice coordinates can be discarded. Note that terms such as R_{xyxy}, $K_{xy,z}$ etc. are referred to the tetrad coordinates.

For a large part of this discussion we will be concerned mainly with the scaling of the coordinates with respect to the typical lattice scale (e.g., to establish that $t = \mathcal{O}(L^2)$). This applies equally well to both coordinate

frames and so, to be specific, we will present the arguments in terms of the tetrad coordinates. Once we have sorted out these scaling issues we will compute the lattice coordinates followed by the tetrad coordinates.

Our first task will be to construct the piece of the Cauchy surface that is covered by a typical computational cell. Recall that we view the Cauchy surface to be a smooth 3-dimensional surface that passes through each vertex of the lattice and that it shares with the lattice, at each vertex, the same future pointing unit normal and second fundamental form (the extrinsic curvatures). In our local Riemann normal coordinates we wish to construct an equation of the form $0 = -t + f(x^u)$ that passes through the vertices of this computational cell and with given extrinsic curvature at the central vertex. For this we use the familiar definition that $\delta n^\mu = -K^\mu{}_\nu \delta x^\nu$ for the small change in the unit normal under a displacement across the Cauchy surface. If we take the displacement to be from the central vertex (o) to a nearby vertex (a) then we have

$$n^\mu_a - n^\mu_o = -K^\mu{}_\nu x^\nu_a \tag{6.1}$$

But we chose the coordinates so that $n^\mu_o = (1,0,0,0)^\mu$ while for the surface $0 = -t + f(x^u)$ the unit normal at (a) is simply $n^\mu_a = g^{\mu\nu}(-1, f_{,u})_\nu / M = (1, f_{,u})^\mu / M$ where $M = 1 + \mathcal{O}(L^2)$ is a normalization factor. Thus we have $(1, f_{,u})^\mu = (1,0,0,0)^\mu - K^\mu{}_\nu x^\nu_a + \mathcal{O}(L^2)$ and this is easily integrated to give

$$t_a = -\frac{1}{2} K_{\mu\nu} x^\mu_a x^\nu_a + \mathcal{O}\left(L^3\right) \tag{6.2}$$

Note that since $K^\mu{}_\nu n^\nu = 0$ we can use this last equation to compute the time coordinates for each vertex in the computational cell (given the spatial coordinates x^u_a and the extrinsic curvatures K_{uv}).

Consider the geodesic segment that joins the central vertex (o) to a typical nearby vertex (a). Then from the definition of Riemann normal coordinates we have

$$x^\mu_a = m^\mu_a L_{oa} \tag{6.3}$$

where m^μ_a is the unit tangent vector to the geodesic at (o).[¶] Thus it follows that

$$|x^\mu_a| = \mathcal{O}(L) \tag{6.4}$$

for each vertex in the computational cell. Combining this with the above equation (6.2) for t_a shows that

$$|t_a| = \mathcal{O}\left(L^2\right) \tag{6.5}$$

[¶]Actually, by virtue of the fact that the path is a geodesic segment expressed in Riemann normal coordinates, the values for m^μ_a are constant along the geodesic.

This result could also be inferred from the simple observation that $m^t \to 0$ as $L \to 0$ (this is a consequence of the smoothness of the Cauchy surface at (o)).

We turn now to the simple question – How accurate do we need the coordinates to be? That is, if $\tilde{x}_i^\mu$ are the exact Riemann normal coordinates for vertex i, then how large can we allow $|x_i^\mu - \tilde{x}_i^\mu|$ to be? The answer can be found by a simple inspection of the evolution equations (3.1–3.2). The truncation terms in those equations are $\mathcal{O}(L^3)$ thus we can safely get by with $\mathcal{O}(L^2)$ errors in the coordinates, that is

$$|x_i^\mu - \tilde{x}_i^\mu| = \mathcal{O}\left(L^2\right) \tag{6.6}$$

The good news is that such coordinates are readily available – flat space will do. To see that this is so, assume, for the moment, that we have estimates for the $K_{\mu\nu}$ and then look back at equations (6.2,1.2). This is a coupled system of equations for the coordinates $(t, x, y, z)_a^\mu$ for each vertex in the computational cell. We are fortunate to have an explicit equation for the time coordinates, namely (6.2). This allows us, in principle, to eliminate each time coordinate that appears in equation (1.2). The result would be a set of equations for the spatial coordinates x_a^u. In the following we will not make this elimination explicit but take it as understood that such a process has been applied. We will have a little more to say on this matter in a short while.

For a typical vertex (l) we will need to compute three spatial coordinates and thus we look to the legs of a tetrahedron. Suppose that that tetrahedron has vertices $(ijkl)$ and suppose that we have computed, by some means, the exact Riemann normal coordinates $\tilde{x}^\mu$ for vertices (ijk). The exact coordinates $\tilde{x}_l^\mu$ for vertex (l) could be obtained by solving the system of equations

$$L_{al}^2 = g_{\mu\nu}(\tilde{x}_a^\mu - \tilde{x}_l^\mu)(\tilde{x}_a^\nu - \tilde{x}_l^\nu) - \frac{1}{3}R_{\mu\alpha\nu\beta}\tilde{x}_a^\mu\tilde{x}_a^\nu\tilde{x}_l^\alpha\tilde{x}_l^\beta \qquad a = i, j, k \tag{6.7}$$

but we could also construct flat space coordinates x_l^μ for vertex l by solving the system

$$L_{al}^2 = g_{\mu\nu}(\tilde{x}_a^\mu - x_l^\mu)(\tilde{x}_a^\nu - x_l^\nu) \qquad a = i, j, k \tag{6.8}$$

From the last equation we conclude that $|\tilde{x}_a^\mu - x_l^\mu| = \mathcal{O}(L)$ for $a \neq l$. Next, make the trivial substitution $\tilde{x}_l^\mu = x_l^\mu + (\tilde{x}_l^\mu - x_l^\mu)$ in the first term in (6.7), expand and use (6.8) to obtain

$$\begin{aligned} 0 = -2g_{\mu\nu}(\tilde{x}_a^\mu - x_l^\mu)(\tilde{x}_l^\nu - x_l^\nu) + g_{\mu\nu}(\tilde{x}_l^\mu - x_l^\mu)(\tilde{x}_l^\nu - x_l^\nu) \\ - \frac{1}{3}R_{\mu\alpha\nu\beta}\tilde{x}_a^\mu\tilde{x}_a^\nu\tilde{x}_l^\alpha\tilde{x}_l^\beta \qquad a = i, j, k \end{aligned}$$

and as each $\tilde{x}^\mu_a = \mathcal{O}(L)$ for $a = i, j, k$ we easily see that

$$|\tilde{x}^u_l - x^u_l| = \mathcal{O}\left(L^3\right) \tag{6.9}$$

The fly in the ointment in the above analysis is the assumption that we knew the $K_{\mu\nu}$ (and thus we could eliminate the t_a). This is not exactly correct for the $K_{\mu\nu}$ are found by solving equations (3.1) which in turn requires the coordinates x^u_a which we have yet to compute (at that stage). Luckily, this is not a major problem. Look carefully at equation (6.7) and recall that $g_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$. Thus the t-terms will appear only in the form $-(\tilde{t}_a - t_l)^2$ and in the curvature terms of the form $R_{tuvw} t_a x^u_a x^v_l x^w_l$. The point to note is that since $t = \mathcal{O}(L^2)$ we see that each of these terms is $\mathcal{O}(L^n)$ with $n \geq 4$ and thus they have no effect on the above analysis. Thus even though we argued previously that we should eliminate the t_a using equation (6.2) the above argument shows that we can put $t_a = 0$ without harm.

Our final calculation concerns the errors induced in t_a by using the approximate x^u_a and $K_{\mu\nu}$ rather than their exact counterparts. Our analysis is very similar to that just presented. We start with the two sets of equations, the approximate and exact equations,

$$2t_a = -K_{uv} x^u_a x^v_a \qquad \text{and} \qquad 2\tilde{t}_a = -\tilde{K}_{uv} \tilde{x}^u_a \tilde{x}^v_a \tag{6.10}$$

We will assume that $|\tilde{K}_{uv} - K_{uv}|$ is at least $\mathcal{O}(L)$ (this is one assumption that we will not relax at a later stage). Then we make the trivial substitution $\tilde{x}^u_l = x^u_l + (\tilde{x}^u_l - x^u_l)$ as above to obtain

$$2\tilde{t}_a = 2t_a - \left(\tilde{K}_{uv} - K_{uv}\right) x^u x^v - 2\tilde{K}_{uv} x^u_a \left(\tilde{x}^v_a - x^v_a\right) - \tilde{K}_{uv} \left(\tilde{x}^u_a - x^u_a\right) \left(\tilde{x}^v_a - x^v_a\right) \tag{6.11}$$

Using $x^u_a = \mathcal{O}(L)$, $\tilde{x}^u_a = \mathcal{O}(L)$ and $|\tilde{K}_{uv} - K_{uv}| = \mathcal{O}(L)$ we find that

$$|\tilde{t}_a - t_a| = \mathcal{O}\left(L^3\right) \tag{6.12}$$

6.1 The lattice coordinates

We return now to the concrete question of how to compute the vertex coordinates within one computational cell. We will first compute the lattice coordinates y^μ followed by the tetrad coordinates x^μ. Our present challenge is to find the solutions of the coupled system of equations

$$L^2_{ab} = g_{\mu\nu}(y^\mu_a - y^\mu_b)(y^\nu_a - y^\nu_b) \tag{6.13}$$

for a suitable subset of the legs (ab) in the computational cell (equal in number to the number of unknown coordinates). The problem here is that if we treat this as a system of equations for the spacetime coordinates $(t,x,y,z)^{\mu}_{a}$ it is extremely unlikely that we will find any solutions (or if we do then the numerics will almost certainly be extremely unstable). The reason is quite simple – the vertices are assumed to lie within one 3-dimensional Cauchy surface. This suggest that we should use the above equations to determine the spatial coordinates $(x,y,z)^{u}_{a}$ with the time coordinates found by other considerations. Fortunately we already know, from the above analysis, that each $|t_a| = \mathcal{O}(L^2)$ while $|y^u_a| = \mathcal{O}(L)$. Thus we see that all terms involving the t_a are $\mathcal{O}(L^4)$ and thus will be consumed by the $\mathcal{O}(L^4)$ truncation errors inherent in the above equation (as an approximation to equation (6.7)). So we may safely discard all the of the t_a terms in the above equations. The next trick that we will use is the observation that the coordinates can be computed one vertex at a time. This is easily shown by direct construction. Consider a typical tetrahedron with vertices $(oijk)$ where (o) is the central vertex and suppose we have computed the coordinates for (ojk). Our task now is to solve the following equations

$$L^2_{ok} = \ g_{uv}y^u_k y^v_k \tag{6.14}$$
$$L^2_{ok} + L^2_{oi} - L^2_{ik} = 2g_{uv}y^u_i y^v_k \tag{6.15}$$
$$L^2_{ok} + L^2_{oj} - L^2_{jk} = 2g_{uv}y^u_j y^v_k \tag{6.16}$$

where the last pair of equations were obtained by expanding $L^2_{ab} = g_{uv}(y^u_a - y^u_b)(y^v_a - y^v_b)$. A simple calculation shows that the solution is given by [4]

$$y^u_k = Py^u_i + Qy^u_j + Rn^u$$

where

$$n^u = g^{uv}\epsilon^{xyz}_{vrs}y^r_i y^s_j$$

$$P = \frac{m_{ik}L^2_{oj} - m_{jk}m_{ij}}{L^2_n} \qquad Q = \frac{m_{jk}L^2_{oi} - m_{ik}m_{ij}}{L^2_n}$$

$$R = \pm\frac{\left(L^2_{ok} - P^2L^2_{oi} - Q^2L^2_{oj} - 2PQm_{ij}\right)^{1/2}}{L_n}$$

$$L^2_n = L^2_{oi}L^2_{oj} - m^2_{ij}$$

and where the m_{ab} are defined by

$$2m_{ij} = L^2_{oi} + L^2_{oj} - L^2_{ij}$$

$$2m_{ik} = L^2_{oi} + L^2_{ok} - L^2_{ik} \qquad 2m_{jk} = L^2_{oj} + L^2_{ok} - L^2_{jk}$$

The two solutions, one for each choice of the $\pm$ sign, correspond to the two possible locations of the third vertex (k), one on each side of the plane containing the triangle (oij). Which choice is taken will depend on the design of the lattice. A systematic choice can be made by noting that the vectors y_i^u, y_j^u and n^u form a right handed system. With $R > 0$ the vector y_k^u lives on the same side of the plane as n^u.

To complete the picture we need coordinates for the first two vertices (1) and (2). Since we chose to align our coordinates so that the x-axis passed through vertex (1) while the vertex (2) is contained in the xy-plane we must have $y_1^u = (A, 0, 0)^u$ and $y_2^u = (B, C, 0)$ for some numbers $A > 0$, B and $C > 0$ such that

$$\begin{aligned} L_{01}^2 &= g_{uv} y_1^u y_1^v \\ L_{02}^2 &= g_{uv} y_2^u y_2^v \\ L_{01}^2 + L_{02}^2 - L_{12}^2 &= 2 g_{uv} y_1^u y_2^v \end{aligned}$$

The solution is readily found to be $A = L_{01}$, $B = (L_{01}^2 + L_{02}^2 - L_{12}^2)/(2L_{01})$ and $C = (L_{02}^2 - B^2)^{1/2}$.

6.2 The tetrad coordinates

The transformation from the lattice to tetrad coordinates is quite simple. Let e_a be the basis for the tetrad frame and let ∂_μ be the corresponding basis for the lattice frame. Recall that we have previously chosen the frames so that both e_1 and ∂_t are aligned with the normal to the Cauchy surface. Now consider a typical vector v_a that joins (0) to (a). In the lattice frame this vector has components y_a^μ while in the tetrad frame, with basis e_b, its components are just $v_a{}^b$. That is we have, for $a = 1, 2, 3$

$$n = \partial_t = e_1 \tag{6.17}$$

$$y_a^t = v_a{}^t \tag{6.18}$$

$$v_a = y_a^\mu \partial_\mu = v_a{}^b e_b \tag{6.19}$$

In the last equation both the y_a^μ and $v_a{}^b$ are known. Thus we have sufficient information to compute ∂_μ in terms of e_a and vice versa. Note that the tetrad coordinates x_a^μ are given by

$$x_a^\mu = v_a{}^b e^\mu{}_b \quad \text{with} \quad e^\mu{}_b = \delta^\mu{}_b \tag{6.20}$$

Finally, using equation (6.2), we can compute the time coordinate for every vertex, not just the three vertices associated with ${v_a}^b$, $a = 1, 2, 3$

$$y^t_a = x^t_a = -\frac{1}{2} K_{\mu\nu} x^\mu_a x^\nu_a \,, \quad a = 1, 2, 3, \cdots \tag{6.21}$$

7 Source terms

We have previously mentioned, without giving details, that source terms such as $R_{xyxy,z}$ can be computed by applying a finite difference approximation to data imported from neighbouring cells. Here we will outline how such a procedure can be applied (the exact details will of course depend on the structure of the lattice). The same procedure can also be used to estimate the spatial derivatives of the ${e^\mu}_a$.

Suppose we have two neighbouring computational cells that have a non-trivial overlap (as indicated in Figure (1)). Each cell will carry values for R_{xyxy} in their own local RNC frames. Our first task would be to import the values form the one cell to the other. This will entail a coordinate transformation, composed of a boost (to account for the change in the unit normal between the two cells) and a spatial rotation (to account for the different orientations of the legs of the cells).

Let x^μ be the (tetrad) coordinates in one cell and let x'^μ be coordinates in the other cell. Our plan is to import data form the x'^μ frame to the x^μ frame. We will demand that the overlap region be such that it contains at least one set of three linearly independent vectors (i.e., legs), at O', which we will denote by w_i, $i = 1, 2, 3$. Since we know the coordinates of each vertex in each cell we can easily compute the components of w_i, $i = 1, 2, 3$ in each frame. The normal vector $n_{o'}$ at O' will have components $n'^\mu_{o'} = (1, 0, 0, 0)^\mu$ in the x'^μ frame. But in the x^μ frame we expect $n^\mu_{o'} = n^\mu_o - {K^\mu}_\nu x^\nu_{o'}$. Thus we have 4 linearly independent vectors at O', expressed in two different frames, and so there must exist a mapping from the components in one frame to those in the other. That is there exists a ${U^\mu}_\nu$ such that

$$n^\mu_{o'} = {U^\mu}_\nu n'^\nu_{o'} \tag{7.1}$$

$$w^\mu_i = {U^\mu}_\nu w'^\nu_i \,, \quad i = 1, 2, 3 \tag{7.2}$$

Since we have values for the components of $n_{o'}$ and w_i, $i = 1, 2, 3$ in both frames we can treat this as a system of equations for the ${U^\mu}_\nu$.

With the ${U^\mu}_\nu$ in hand, we can compute the values of $R_{\mu\alpha\nu\beta}$ at O' in the x^μ frame of O by way of

$$\left(R_{\mu\alpha\nu\beta}\right)_{o'} = {U_\mu}^\theta {U_\nu}^\phi {U_\alpha}^\rho {U_\beta}^\tau \left(R'_{\theta\phi\rho\tau}\right)_{o'} \tag{7.3}$$

with ${U_\mu}^\nu = g_{\mu\alpha} g^{\nu\beta} {U^\alpha}_\beta$ and $g_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$. This can be repeated for all of the vertices that surround O. The result is a set of point estimates for $R_{\mu\alpha\nu\beta}$ in the neighbourhood of O which in turn can be used to estimate the derivatives of $R_{\mu\alpha\nu\beta}$ at O. This part of the process is similar to that required when computing the Hessian (see below) and presumably similar methods could be applied.

Note that for a sufficiently refined lattice, the ${U^\mu}_\nu$ should be close to the identity map, that is ${U^\mu}_\nu = {\delta^\mu}_\nu + {V^\mu}_\nu \mathcal{O}(L)$ where the ${V^\mu}_\nu$ are each of order $\mathcal{O}(1)$. This can be used to simplify some of the above computations.

See [1] for a complete example in the context of the Schwarzschild spacetime.

In section (5.2) we noted that substitutions such as $R_{tyxy} = \mathcal{R}_{abcd} {e_t}^a {e_y}^b {e_x}^c {e_y}^d$ could be introduced into the curvature evolution equation (5.13). At that time we argued that that was not necessary for the coordinate data, in this instance R_{tyxy}, could easily be recovered when needed by using $R_{tyxy} = \mathcal{R}_{abcd} {e_t}^a {e_y}^b {e_x}^c {e_y}^d$. Then the scheme described above could be used to compute $R_{tyxy,x}$. However there may be numerical advantages in making a formal substitution before estimating any of the partial derivatives. For $R_{tyxy,x}$ this would lead to the following

$$\begin{aligned} R_{tyxy,x} &= \left(\mathcal{R}_{abcd} {e_t}^a {e_y}^b {e_x}^c {e_y}^d\right)_{,x} \\ &= \mathcal{R}_{abcd,x} {e_t}^a {e_y}^b {e_x}^c {e_y}^d + \mathcal{R}_{abcd} \left({e_t}^a {e_y}^b {e_x}^c {e_y}^d\right)_{,x} \end{aligned}$$

Since the $\mathcal{R}_{abcd}$ are scalars, their partial derivatives can be estimated without requiring any of the frame transformations described above (importing such data from neighbouring cells is trivial). This leaves us with the derivatives of the form $({e_\mu}^a)_{,x}$. Since $n_\mu = -{e_\mu}^1$ we can use (5.4) to eliminate any of the spatial derivatives of ${e_\mu}^1$, in this case $({e_\mu}^1)_{,x}$. This would introduce the extrinsic curvatures into the evolution equations. However the remaining partial derivatives, $({e_\mu}^i)_{,x}$, $i = 2, 3, 4$, would have to be estimated using the methods described above (by importing data from neighbouring cells etc.). This approach does incur a small computational overhead which may be justified if it brings some improvement to the quality of the numerical data (e.g., better accuracy and or stability). Judging the merits of this variation against the simple method given in section (5.2) might best be decided by direct numerical experimentation.

7.1 Extrinsic curvatures

A cursory glance at equation (3.1) might give the impression that it constitutes a simple linear system for the K_{uv}. But things are never as simple as they seem. The problem, as already noted, is that there are far too many equations for the six K_{uv}. If we make the reasonable assumption that the lattice data is a good approximation to the (unknown) continuum spacetime then we can expect considerable redundancy in this overdetermined system. How then do we pull out just six equations for the six K_{uv}? One option is to reject all but six of the equations and hope that this yields an invertible system for the K_{uv}. A better, and more flexible approach, is to take a weighted sum of the equations, that is we create a new set of equations of the form

$$0 = \sum_{ab} W^n_{ab} \left(P_{ab} - K_{uv} \Delta x^u_{ab} \Delta x^v_{ab} \right) \tag{7.4}$$

where W^n_{ab} are a set of weights of our own choosing (typical values being 0 and ± 1). With $n = 1, 2, 3 \ldots 6$ we have six equations for the six unknowns. This idea has been used previously [4] and worked very well. There are certainly other options that could be explored (e.g., different choices of weights, least squares) but we have tested none simply because the above scheme seems to work well.

7.2 The Hessian

At some point we will need to estimate the $N_{|uv}$ at a central vertex. Since N is a scalar function and since we are using Riemann normal coordinates this computation is essentially that of computing all of the second partial derivatives on an unstructured grid. There is an extensive literature on this point in the context of finite element schemes. We mention here one approach which we discussed in one of our earlier papers [4] (but which we have yet to test).

Consider a typical leg (ij) in some computational cell. We can estimate $N_{|u}$ at the centre of the leg by the centred finite difference approximation

$$\left(N_{|u} \right)_{ij} = \frac{N_j - N_i}{L_{ij}} \left(m_u \right)_{ij} \tag{7.5}$$

in which $(m_u)_{ij}$ is the unit vector tangent to the geodesic and oriented so that it points from (i) to (j). We can repeat this computation for each leg

in the computational cell and then estimate $N_{|uv}$ by a least squares fit of the function

$$\tilde{N}_{|u}(x) = \tilde{N}_{|u} + \tilde{N}_{|uv} x^v \tag{7.6}$$

to the data generated above by equation (7.5). A suitable least squares sum would be

$$S(\tilde{N}_{|u}, \tilde{N}_{|uv}) = \sum_u \sum_{ij} \left(\left(N_{|u}\right)_{ij} - \tilde{N}_{|u} - \tilde{N}_{|uv}\, \bar{x}^v_{ij} \right)^2 \tag{7.7}$$

where $\bar{x}^v_{ij}$ is the centre of the leg (ij). Note that this least squares fit must be made subject to the constraint $N_{|uv} = N_{|vu}$. The coefficients $\tilde{N}_{|u}$ and $\tilde{N}_{|uv}$ would then be taken as our estimates for the corresponding quantities at the central vertex.

8 Discussion

There are a number of aspects of this paper that could easily be debated. For example, should we proceed with the substitutions such as $R_{tyxy} = \mathcal{R}_{abcd} e_t{}^a e_y{}^b e_x{}^c e_y{}^d$ in equation (5.13)? As already noted in section (7) this would introduce a raft of new terms including the extrinsic curvatures. We chose not to use the substitution solely for reasons of simplicity. There is also a question over our choice of tetrad. Do we really need to demand that the tetrad be orthonormal? Not at all. We could choose to tie the tetrad to the legs of the lattice (and then the tetrad would no longer be needed) but that would produce a coupling amongst all of the evolution equations (e.g., the evolution equation for $\mathcal{R}_{xyxy}$ would be a linear combination of all of the evolution equations for $R_{\mu\alpha\nu\beta}$). The resulting equations would not be anywhere near as simple as those listed in the Appendix. Then we have the issue of estimating partial derivatives on an irregular lattice (for the Hessian and the source terms in the curvature evolution equations). This is non-trivial but at least there is an extensive literature on the subject and so a workable solution should not be too hard to find (which may be the least squares method suggested in section (7.2)). All of these issues (and most likely others) can be explored by direct numerical exploration on a non-trivial $3+1$ spacetime. We plan to report on such investigations soon. For a simple application to the $1+1$ Schwarzschild spacetime see [1].

A The curvature evolution equations

Here we list all 14 curvature evolution equations (this follows on from section (5.2) where we provided details of the derivation for the first equation below).

$$\frac{d\mathcal{R}_{xyxy}}{dt'} = R_{tyxy,x} - R_{txxy,y} + \mathcal{R}_{tyxy}\nabla_x N - \mathcal{R}_{txxy}\nabla_y N + \mathcal{R}_{tyxy}\nabla_x N - \mathcal{R}_{txxy}\nabla_y N \tag{A.1}$$

$$\frac{d\mathcal{R}_{xyxz}}{dt'} = R_{tzxy,x} - R_{txxy,z} + \mathcal{R}_{tyxz}\nabla_x N - \mathcal{R}_{txxz}\nabla_y N + \mathcal{R}_{tzxy}\nabla_x N - \mathcal{R}_{txxy}\nabla_z N \tag{A.2}$$

$$\frac{d\mathcal{R}_{xyyz}}{dt'} = R_{tzxy,y} - R_{tyxy,z} + \mathcal{R}_{tyyz}\nabla_x N - \mathcal{R}_{txyz}\nabla_y N + \mathcal{R}_{tzxy}\nabla_y N - \mathcal{R}_{tyxy}\nabla_z N \tag{A.3}$$

$$\frac{d\mathcal{R}_{xzxz}}{dt'} = R_{tzxz,x} - R_{txxz,z} + \mathcal{R}_{tzxz}\nabla_x N - \mathcal{R}_{txxz}\nabla_z N + \mathcal{R}_{tzxz}\nabla_x N - \mathcal{R}_{txxz}\nabla_z N \tag{A.4}$$

$$\frac{d\mathcal{R}_{xzyz}}{dt'} = R_{tzxz,y} - R_{tyxz,z} + \mathcal{R}_{tzyz}\nabla_x N - \mathcal{R}_{txyz}\nabla_z N + \mathcal{R}_{tzxz}\nabla_y N - \mathcal{R}_{tyxz}\nabla_z N \tag{A.5}$$

$$\frac{d\mathcal{R}_{yzyz}}{dt'} = R_{tzyz,y} - R_{tyyz,z} + \mathcal{R}_{tzyz}\nabla_y N - \mathcal{R}_{tyyz}\nabla_z N + \mathcal{R}_{tzyz}\nabla_y N - \mathcal{R}_{tyyz}\nabla_z N \tag{A.6}$$

$$\frac{d\mathcal{R}_{tyxy}}{dt'} = R_{xyxy,x} - R_{xyyz,z} + \mathcal{R}_{iyxy}\nabla^i N + \mathcal{R}_{tyty}\nabla_x N - \mathcal{R}_{txty}\nabla_y N \tag{A.7}$$

$$\frac{d\mathcal{R}_{txxy}}{dt'} = -R_{xyxy,y} - R_{xyxz,z} + \mathcal{R}_{ixxy}\nabla^i N + \mathcal{R}_{txty}\nabla_x N - \mathcal{R}_{txtx}\nabla_y N \tag{A.8}$$

$$\frac{d\mathcal{R}_{tzxy}}{dt'} = R_{xyxz,x} + R_{xyyz,y} + \mathcal{R}_{izxy}\nabla^i N + \mathcal{R}_{tytz}\nabla_x N - \mathcal{R}_{txtz}\nabla_y N \tag{A.9}$$

$$\frac{d\mathcal{R}_{tzxz}}{dt'} = R_{xzxz,x} + R_{xzyz,y} + \mathcal{R}_{izxz}\nabla^i N + \mathcal{R}_{tztz}\nabla_x N - \mathcal{R}_{txtz}\nabla_z N \tag{A.10}$$

$$\frac{d\mathcal{R}_{txxz}}{dt'} = -R_{xyxz,y} - R_{xzxz,z} + \mathcal{R}_{ixxz}\nabla^i N + \mathcal{R}_{txtz}\nabla_x N - \mathcal{R}_{txtx}\nabla_z N \tag{A.11}$$

$$\frac{d\mathcal{R}_{tyxz}}{dt'} = R_{xzxz,x} - R_{xzyz,z} + \mathcal{R}_{iyxz}\nabla^i N + \mathcal{R}_{tytz}\nabla_x N - \mathcal{R}_{txty}\nabla_z N \tag{A.12}$$

$$\frac{d\mathcal{R}_{tzyz}}{dt'} = R_{xzyz,x} + R_{yzyz,y} + \mathcal{R}_{izyz}\nabla^i N + \mathcal{R}_{tztz}\nabla_y N - \mathcal{R}_{tytz}\nabla_z N \tag{A.13}$$

$$\frac{d\mathcal{R}_{tyyz}}{dt'} = R_{xyyz,x} - R_{yzyz,z} + \mathcal{R}_{iyyz}\nabla^i N + \mathcal{R}_{tytz}\nabla_y N - \mathcal{R}_{tyty}\nabla_z N \tag{A.14}$$

Note that in the above there are two instances of $\mathcal{R}_{txyz}$, in (A.3) and (A.5), and these should be replaced with $\mathcal{R}_{tyxz} - \mathcal{R}_{tzxy}$.

B Riemann normal coordinates

We recall here a few basic properties of Riemann normal coordinates. A set of coordinates x^μ are said to be in Riemann normal form if every geodesic passing through a given point O (the origin) is described by $x^\mu(s) = sv^\mu$ where s is an affine parameter and v^μ is constant along the geodesic. It follows from the geodesic equation and its successive derivatives, that the connection and its higher symmetric derivatives[‖] all vanish at the chosen point, that is at O

$$0 = \Gamma^\mu_{\alpha_1\alpha_2} \tag{B.1}$$

$$0 = \Gamma^\mu_{(\alpha_1\alpha_2;\alpha_3\cdots\alpha_n)} \qquad n = 3, 4, 5, \cdots \tag{B.2}$$

These conditions do not uniquely determine the coordinates for we are free to apply a transformation of the form $x^\mu \mapsto \Lambda^\mu{}_\nu x^\nu$ which clearly preserves the property that the geodesics through O are of the form $x^\mu(s) = sv^\mu$. This freedom can be used to ensure that the metric at O is simply $g_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$.

Choosing the coordinates so that the connection vanishes at the origin does introduce some nice properties, in particular covariant differentiation reduces,

[‖]Here we take a small liberty with notation, the upper index on the Christoffel symbol should be ignored when computing covariant derivatives.

at the origin, to simple partial differentiation. This fact was essential to the analysis given in sections (4).

There are two main impediments to the existence of Riemann normal coordinates. The metric must be smooth throughout the neighbourhood (i.e., away from curvature singularities) and each point in the neighbourhood should be connected to the origin by exactly one geodesic (i.e., no pair of geodesics through O should cross, except at O). These conditions are easily satisfied by simply choosing the neighbourhood around O to be sufficiently small (but not vanishingly small).

In these coordinates the metric and connection can be expanded as a Taylor series around O leading to

$$g_{\mu\nu}(x) = g_{\mu\nu} - \frac{1}{3}R_{\mu\alpha\nu\beta}x^\alpha x^\beta - \frac{1}{6}R_{\mu\alpha\nu\beta,\gamma}x^\alpha x^\beta x^\gamma + \mathcal{O}\left(L^4\right) \tag{B.3}$$

$$g^{\mu\nu}(x) = g^{\mu\nu} + \frac{1}{3}R^\mu{}_\alpha{}^\nu{}_\beta x^\alpha x^\beta + \frac{1}{6}R^\mu{}_\alpha{}^\nu{}_{\beta,\gamma}x^\alpha x^\beta x^\gamma + \mathcal{O}\left(L^4\right) \tag{B.4}$$

$$\begin{aligned}\Gamma^\mu_{\alpha\beta}(x) = {} & \frac{1}{3}R^\mu{}_{\alpha\gamma\beta}x^\gamma + \frac{1}{24}\left(2R^\mu{}_{\gamma\delta\beta,\alpha} + 4R^\mu{}_{\alpha\delta\beta,\gamma} + R_{\gamma\alpha\delta\beta}{}^{,\mu}\right)x^\gamma x^\delta \\ & + (\alpha \leftrightarrow \beta) + \mathcal{O}\left(L^3\right)\end{aligned} \tag{B.5}$$

If we know the Riemann normal coordinates, x^μ_i and x^μ_j, for a pair of points, i and j, then we can compute the length of the geodesic segment that joins the points by

$$L^2_{ij} = \left(g_{\mu\nu} - \frac{1}{3}R_{\mu\alpha\nu\beta}\bar{x}^\alpha_{ij}\bar{x}^\beta_{ij} - \frac{1}{6}R_{\mu\alpha\nu\beta,\gamma}\bar{x}^\alpha_{ij}\bar{x}^\beta_{ij}\bar{x}^\gamma_{ij}\right)\Delta x^\mu_{ij}\Delta x^\nu_{ij} + \mathcal{O}\left(L^6\right) \tag{B.6}$$

where $\Delta x^\mu_{ij} := x^\mu_j - x^\mu_i$ and $\bar{x}^\mu_{ij} := (x^\mu_j + x^\mu_i)/2$ is the mid-point of the leg. The unit tangent vector m^μ_{ij} to the geodesic at i, is given by

$$\begin{aligned}L_{ij}m^\mu_{ij} = {} & \Delta x^\mu_{ij} + \frac{1}{3}x^\alpha\Delta x^\nu_{ij}\Delta x^\beta_{ij}R^\mu{}_{\nu\alpha\beta} + \frac{1}{12}x^\alpha x^\nu\Delta x^\beta_{ij}\Delta x^\gamma_{ij}R^\mu{}_{\alpha\nu\gamma,\beta} \\ & + \frac{1}{6}x^\alpha x^\nu\Delta x^\beta_{ij}\Delta x^\gamma_{ij}R^\mu{}_{\beta\nu\gamma,\alpha} + \frac{1}{24}x^\alpha x^\nu\Delta x^\beta_{ij}\Delta x^\gamma_{ij}R^{,\mu}_{\alpha\beta\nu\gamma} \\ & + \frac{1}{12}x^\alpha\Delta x^\nu_{ij}\Delta x^\beta_{ij}\Delta x^\gamma_{ij}R^\mu{}_{\beta\alpha\gamma,\nu}\end{aligned} \tag{B.7}$$

Finally, if we have a geodesic triangle built on the three points i, j, k then the generalised cosine law takes the form

$$2L_{ik}L_{jk}\cos\theta_{ij} = L^2_{ik} + L^2_{jk} - L^2_{ij} - \frac{1}{3}R_{\mu\alpha\nu\beta}\Delta x^\mu_{ik}\Delta x^\nu_{ik}\Delta x^\alpha_{jk}\Delta x^\beta_{jk} + \mathcal{O}\left(L^5\right) \tag{B.8}$$

in which θ_{ij} is the angle subtended at vertex k by the geodesic that connects i to j.

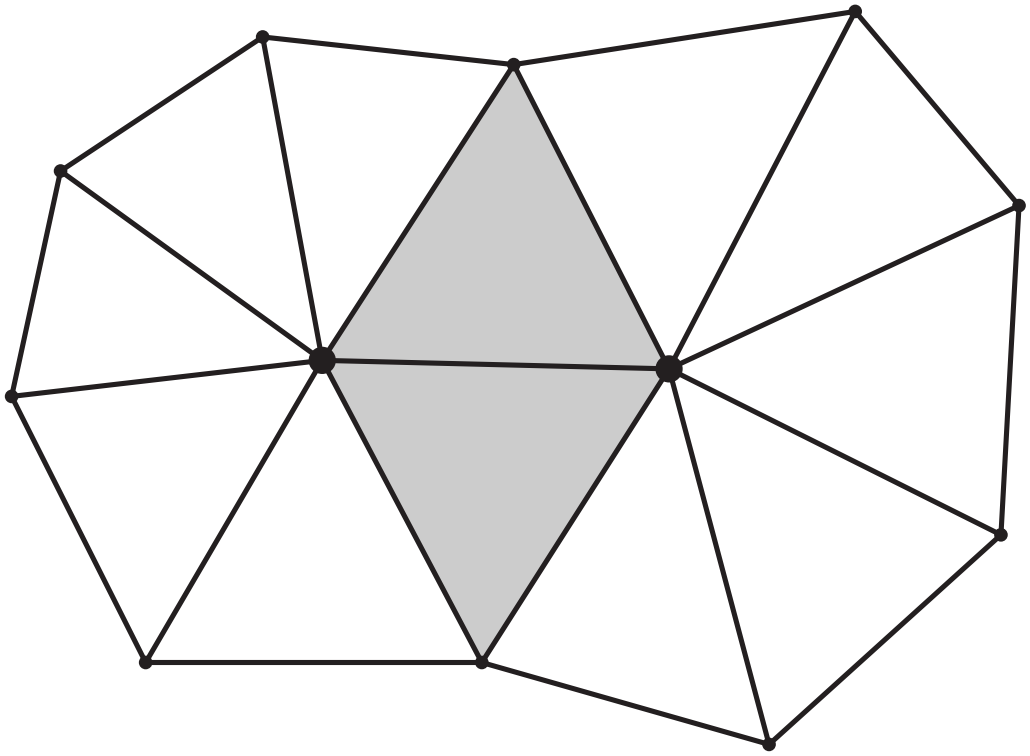

Figure 1: An example of the overlap, the shaded region, between a pair of computational cells. The central vertex of each computational cell is denoted by the large dots whereas the smaller dotes denote the vertices that define the boundary of the computation cells. These vertices are themselves the central vertices of other computational cells. In this 2-dimensional example the overlap consists of just the pair of triangles. In 3 dimensions the over lap would consist of a closed loop of tetrahedra. In each case there is ample information available to obtain a coordinate transformation between the pair of local Riemann normal frames.

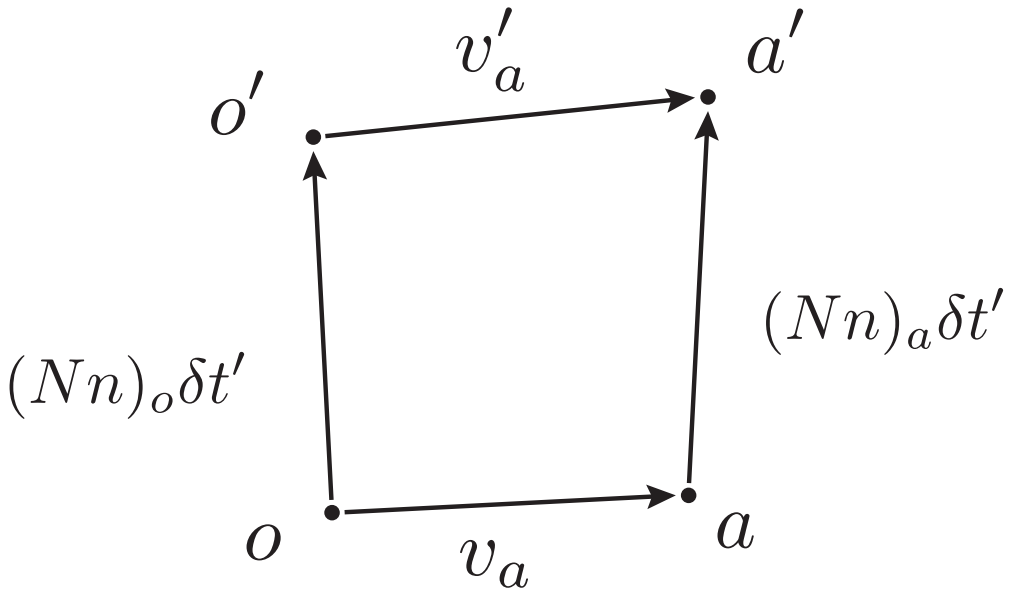

Figure 2: Here we show the evolution of one leg (oa) within one computational cell. Clearly the four vectors form a closed loop and thus $(Nn)_o\delta t' + v'_a = v_a + (Nn)_a)\delta t'$ which leads directly to equation (5.6).

References

[1] L. Brewin, An Einstein-Bianchi system for Smooth Lattice General Relativity. I. The Schwarzschild spacetime., arXiv:1101.3171.

[2] L. Brewin, Long term stable integration of a maximally sliced Schwarzschild black hole using a smooth lattice method, *Class. Quantum Grav.* **19** (2002) 429–455.

[3] L. Brewin and J. Kajtar, A Smooth Lattice construction of the Oppenheimer-Snyder spacetime, *Phys. Rev. D* **80** (2009) 104004, arXiv:0903.5367. http://users.monash.edu.au/~leo/research/papers/files/lcb09-05.html.

[4] L. Brewin, An ADM 3+1 formulation for smooth lattice general relativity, *Class. Quantum Grav.* **15** (1998) 2427–2449.

[5] L. Brewin, Riemann normal coordinates, smooth lattices and numerical relativity, *Class. Quantum Grav.* **15** (1998) 3085–3120.

[6] L. Brewin, Riemann Normal Coordinate expansions using Cadabra, *Class. Quantum Grav.* **26** (2009) 175017, arXiv:0903.2087. http://users.monash.edu.au/~leo/research/papers/files/lcb09-03.html.

[7] L. Brewin, Deriving the ADM 3+1 evolution equations from the second variation of arc length, *Phys. Rev. D* **80** (2009) 084030, arXiv:0903.5365. http://users.monash.edu.au/~leo/research/papers/files/lcb09-04.html.